Simulating Real World Evolutionary Phenomena

Thomas Frederick Turton

BSc in Computer Science

27/04/2016

SCHOOL OF COMPUTING AND MATHEMATICS
Keele University
Keele
Staffordshire
ST5 5BG

Abstract:

The idea of simulated virtual evolution has been around for a long time, however the use of evolutionary algorithms has dominated the ideas of the field. Creating a more open and agent controlled method of simulating evolution is essential to the creation of more realistic evolution simulations. This project attempts to provide a simple and more agent controlled system of simulating real world evolution. These agent controlled evolution systems branch from ideas within the natural world and other studies done in the same area. Results from this project provide evidence that to at least a certain extent there was a lot of success, however there is much to build upon to improve the realism and reliability of such simulations.

1 - Introduction:

In this section I will discuss what this project is trying to achieve and what the reason behind this project is. Furthermore the underlying concepts and ideas that have formed the foundation of this project will be discuss and explained. Many of these ideas and concepts have been built upon ideas stated in previous studies. Other ideas in this project have come directly from real life observations. There will also be a discussion about what the expected outcomes of this project will be.

1.1 - What is this project?

This project asks whether or not it is possible to create an increasingly complex evolution of agents, by simply adding a series of variables that determine each agent's characteristics. These characteristics determine everything each agent does in the 3D world ranging from their visual size and colour (i.e. physical characteristics) to the way they choose where to move dependent on the positions of other species (i.e. their behaviour).

Unlike many other simulations that tackle evolving 3D agents (Harvey 1992 see also, Stanton and Channon, 2015) no evolutionary algorithm will be employed, at least none in the form most commonly seen (for example the algorithms seen in the work of Sims, 1994; Miconi and Channon 2005). The reason that a evolutionary algorithm will not be used was to prevent the idea that an outside control is choosing which of the individuals are the best to be used to create the new generation. Frequently in real world evolution the incorrect path will be taken and end up creating interesting results, this method of evolution should help to allow these interesting results occur. This should therefore lead to a similar result as that within the real world, as not all reproductive systems will choose their mate via what is deemed to be the most suitable. Instead of an evolutionary algorithm, evolution will be decided by more simple mechanisms. By using the idea of the survival of the fittest placed down by Charles Darwin within On the Origin of Species (as edited by G. Beer 2008), the simulations will decided on the 'fittest' individuals quite simply by those agents which survive long enough to pass on their genetic data. Since the 'fittest' individuals are decided by those that survive at no point are all the individuals compared to each other as seen in evolutionary algorithms.

1.2 - Reason for this project?

This project aims to provide a visually appealing and easily understood simulation to show how evolution works. Thus it could potentially be used as an educational tool which may help to introduce the idea of evolution and in addition introduce concepts that can be transferred into real world examples. Furthermore, this project should also provide a base to improve upon in order to create a more advanced simulation, which would be able factor in more complex behaviours, characteristics and situations such as parental instincts and symbiosis. This more advanced simulation could then introduce a higher level of understand of evolution as well as introduce people to more complex ideas about evolution.

1.3 - Evolution

Evolution has been defined as being the changes which occur between descendants of a species due to heritable traits over successive generations therefore altering biological populations over time (Strickberger, Hall, and Hallgrimsson, 2007). Evolution is required component in ecological systems, which are systems the deal with relations of organisms and each other or the environment. Mutation is also a very vital component in evolution as it allow the traits within a set of descendants to change. Without mutation as a factor within evolution the changes between each generation will be none existent and so no alterations will be made to the population.

1.4 - Predicted Phenomena

Since the ideas that will be implemented have been designed with ideas from nature in mind, it is possible to make predictions on the behavior that could possibly occur. Furthermore if this simulation is to give an accurate example of real life evolution it should display phenomena that occur in real life evolution. Whether or not these behaviors present themselves depends on the chance they have of occurring, whether or not they were implemented correctly and if the ideas implemented were correct. Implementing a simulation which allows for these phenomena to occur will be a difficult process as outlined by Costa et al., (2011) stating that “modeling and simulation of ecological systems are major challenges”. These phenomena can be broken up into two main sections, Intraspecific and Interspecific. The phenomena will be split between the two depending on if the phenomena happens within a species (Intraspecific) or between species (Interspecific).

Intraspecific

A Dictionary of Biology (Martin and Hine, 2008, p. 315) defines Intraspecific as “Occurring between members of the same species”. Examples of intraspecific phenomena that could happen and possibly been seen in the simulation are as follows.

- Flocking

Since individuals within a species will not consider each other a viable option for food, at least not until dire circumstances, they should stay around each other as a form of safety. However species that are more likely considered predators rather than prey may show a weaker version of this with less of a need to find safety in numbers. An example of flocking would be swift murmurations which can be seen when high amounts of swift birds fly in a single area. The murmurations are created by the flocking of the birds towards each other. The birds have a need to be near birds of the same species and so keep flying together.

- Self selected traits

Due to certain traits being dependent on others, as well as certain traits being more important in certain situations, it is natural to assume that the more preferable combination of traits will selected out of the populations in order to increase the fitness of the individuals. Furthermore since only the fittest individuals survive long enough to mate, only the fittest traits will survive and be passed down through the generations.

- Migrating Populations

Following on from the previous phenomenon it can be expected that since the agents decide on their own ‘fittest’ traits that the most common traits of the population will change over time. These changing common traits can be described as migrating as the population is moving around in the fitness space.

Interspecific

A Dictionary of Biology(Martin and Hine, 2008, p. 315) defines Interspecific as “Occurring between members of different species”. Examples of interspecific phenomena that should been seen in the simulation are as follows.

- Divergent Evolution

“Divergent selection may arise from external differences between the two environments or niches”(Rundle and Schluter as in Diekmann, *et al*, 2004, p.193). The act of a single population of the same species moving apart to the point of being considered two separate species. This phenomenon should be seen since the ever mutating genotype of each agent will frequently cause differences in each species. If these differences occur in a large enough group of individuals then they will begin to branch away from the rest of the species. Over time this branched subspecies will become different enough to be considered its own unique species. Much like the example of Darwin's finches where one breed of finch branched into many different subspecies each with notable difference (Lack, 1983).

- Convergent Evolution

Convergence in real world evolution is the act of two separate species developing similar attributes due to being in similar environments. As discussed by Jonathan Losos(2011), who states that convergent evolution is "the independent evolution of similar features in different evolutionary lineages"(p. 1827). These ideas were further research by David Stern(2013) who stated that "Convergence often results from similar genetic changes"(p. 751). Although not completely like the real world phenomenon, the simulation phenomenon will differ due to the simplicity of the agents genotypes. This simplicity means that the convergence that should be seen can be defined as the act of two different populations genotypes becoming so like each other that they consider each other part of the same species in terms of mating and aggression. Due to the method of deciding species, when two individuals from two separate species encounter each other, if they have a similar enough genotype will then consider each other as being the same species. This may lead to three possibilities which may occur. These possibilities however, are isolated events and only one possibility can occur each time convergent evolution happens. The first of these is when the two species start to interbreed they could become a single new species. Secondly, one of the species may remain unaffected and live alongside the new mixed species. The final possibility is that both the starting species and the mixed species will remain present.

- Predator-Prey dynamics (Kuang and Beretta, 1998)

This is a dynamic in which two species population sizes are dependent on one another (i.e. need each other in order to survive). The predator would be the stronger, or more aggressive of the two agents, and is dependent on the prey as a food source. Therefore, if the population of the prey decreases to such a dangerous amount the predators population with follow suit. Furthermore if the population of the predators decreases, then the population of the prey should increase. This dynamic is very apparent in nature and has been studied many times and as such if my simulation is an apt example of nature then it should be seen.

2 - Methodology:

In this chapter the technology and programs used to create the simulation will be covered as well as all the design principles and methods this project will use to be implemented. Much of the decisions made in the chapter will greatly help the later stages of the project and as such provide much of the foundation of this project.

2.1 - Technology used

Although for the implementation of this project a great different amount of software could have been used each of the software chosen was done so for good reason which will be discussed in this section.

Unreal Engine 4

Unreal Engine 4 was used as the platform and game engine to implement this project as it provided a very useful platform in order to build my simulation. This is because it has certain important features that are pre made within the platform. One such feature is the premade simple AI system that was built upon for this project. This premade AI system prevented the need to create the lower level control of the agents, such as where they could move in the 3D environment. This allowed the focus to be moved onto the more important aspects of the project, such as the evolved behavior of the agents. Specifically the inbuilt AI navigation allowed the simulation to be in a more realistic 3D environment with hills and mountains rather than the environment being a simple flat plane. This more realistic environment helps in that it provides the viewer with a more interesting and visually appealing environment. Using the behavior tree system within Unreal Engine 4 has allowed the agents to have a viewable process of making a decision. Finally Unreal Engine 4 will allow for the final finished simulation to be exported as a single executable file which will allow it to increase its ease of use.

Visual Studios

Visual Studios was used as the Integrated Development Environment (IDE) to write the C++ code that was required. Visual Studios was used in this project as it was the IDE that Unreal Engine 4 uses by default although it also provided a very useable platform allowing the C++ code to be written with ease. In addition the fact that Visual Studios was the default IDE for Unreal Engine 4 meant that the transition and moving of code from one to the other was made easier than it would with another IDE.

Google Sheets

Google Sheets will be used as the spreadsheet software in this project and as such will be used to analyse the data. The analysing of the data also involves the creation of any needed graphs of data. The base .txt files that the simulation will output will be taken into Google Sheets and translated into tables of raw data which will then be used in order to create the necessary graphs. Although there may be better choices of spreadsheet software to use this software was chosen as it is free and very simple to use.

2.2 - Prototyping

Use of a prototyping methodology was useful in the creation of this project as it helped to stagger the complexity of simulation. Due to the large complexity predicted in the final version for this project it was required to split up the needed components into smaller less complex versions. This method also meant that in the case that the any of the versions proving to be too complex and take more time than available for this project a previous version was still usable. Furthermore this methodology help to make each version simpler than would be if implemented singularly since much of the implementation was done in a previous version.

Version 1

This version is built to only show a basic population of pawns with the criteria that a big size is fitter. As such this version will not show many complex interactions other than the need to reproduce. Furthermore the movement of this versions pawns will be mostly based on choosing a random point within its own area. However when needing to reproduce the pawn will go towards another pawn. Since no method to determine species has been implemented the whole population of pawns will be considered one species. Although comparatively simple to the future versions this version does allow for the basic understanding that evolution occurs through the passing of stronger genes as well as the addition of these genes.

- No colour
- No health just lifespan
- Simple random movement

- No determination between species

Version 2

Version 2's most important factor is the addition of species determination between every individual which is done simply through the comparison of hue values. With the addition of dynamic species determination more decisive movement can be implemented as well as populations of the same species deciding on their own fittest traits. In this version more interesting behaviour and phenomena can finally occur and a better understanding of evolution can be taught using it.

- Colour
- Species determination
- Decisive movement

Version 3

Version 3 mainly improves upon version 2 by adding a more visually appealing simulation. The main change is the addition of more complex agent models with animations specific to what state they are at that point of time. Furthermore this version will continue to cement the difference between a predator or prey by the agents model being decided by their genotypes. The three main versions of creature that will be implemented will be based on real life animals. Firstly, the fast and aggressive predator agent based on a lion. Secondly, the slow, large and passive agent based on an elephant. Thirdly, the small and quickly reproducing prey agent based on a rabbit. All three of these agent types should cover the kinds of agents that may develop in the environment however a generic creature that does not fall into any of the three versions will also be implemented. Finally there will be an addition of a source of food which is not another pawn and will not move. The addition of this food source will help to promote the occurrence of prey like agents which in turn will make the predator-prey phenomenon more visible and frequent. This food will also mean that in the event that a single species outcompetes all the other species to extinction they can still survive as long as they develop to the new food source.

- Specific models to genotype
- Full Genotype Species Check
- State animations
- Addition of non-agent food source

3 - Design:

Before even implementing the features needed to run the simulation the structure and use of the features had to be determined and then designed. In this section each of the features will be talked about in terms of how and why they will be added. In addition to the features that will be added to the simulation the test cases that will be used to determine how successful or unsuccessful this simulation is will also be outlined.

3.1 - Agents

The agents consist of three main components, the Pawn Blueprints Class which derives from the base C++ class, the Controller Blueprints Class which contains all the information needed to have interactions between agents as well as the AI functionality, and the Behaviour Tree. The Behaviour Tree contains the decision methods which will determine the actions that the agents could possibly take. When the version two agents are being created a very important value must be added to the genotype of all of the agents which will be called the Species Variance Acceptance(SVA). The species determination method added in agent version two will be achieved with the addition of this value (SVA) which will be used as the acceptable difference between two individual agents colour values. If the difference between the agents colour is larger than the SVA then they will be considered two different species. This method of implementing species will help to create very dynamic species boundaries. However beneficial this method of creating species is for this simulation it also comes with a very large problems with the speed at which the simulation can run. Due to the constant need for checks that the agents must make, the amount of processing that will occur during the program will exponentially increase with the size of the total population.

3.2 - GUI Elements

In order to provide a level of feedback to the viewer of the simulation simple Graphic User Interface (GUI) elements had to be employed. Although simple, these elements still provided all the information and control the viewer would require. The GUI includes a main menu, pause and fastforward buttons as well as a details panel for each of the agents. The details panel will be accessed by the viewer when pressing on an agent. This will allow the viewer of the simulation to see the information that is present about each agent due to the fact that this panel will contain all of the agent's genotype data.

3.3 - Saving Data

In order to do any quantitative data analysis using data from the simulation it is required that data is exported out of Unreal and saved in a readable file format. The process of saving the data within the genotypes of the agents will require an loop through the entire population recording all of the values needed and exporting this data.

3.4 - Test Cases

The data that is required from this simulation to determine how successful it has been can be broken into two sections, saved data values and visual data. Each and every agent within the simulation has the same selection of values within their genotype however these values will change. These changes in the genotype values can be saved and exported to a spreadsheet program and analysed. Comparatively certain phenomena are much harder to recognise by the data values in each run, or at least a majority of the runs. To establish this it will be imperative to record the runs and then view if and when these phenomena happen visually.

Data values of the genotypes will be saved every five seconds to ensure a large amount of data will be collected and the data is a good interpretation of the simulation. I will have ten run throughs of the simulation and take data from all of the agents within the simulation at that time. This collected data will be analysed in multiple ways, as follows:

- Plotting size of agents over time (with max and min of population)

This test case will help to see whether or not the simulation has just created agents that increase or decrease in size over time or if it is more dynamic and chosen by the specific populations. In order for this simulation to be a correct interpretation of real life evolution the size should both decrease and increase over time depending on the situation the agents find themselves in and not just simply increase.

- Graph showing clusters of similar coloured (species) agents at different times.

This test case will provide information on whether or not the species are migrating to specific species and creating specific populations. Furthermore, this data should help to show that over time the populations may change instead of just finding a selection of values and sticking to them.

- Percentage of population with Hue values.

This test should determine if the normal distribution curve can be seen in populations within this simulation. The normal distribution curve shows that there is a decrease in percent of population having a certain trait further from the norm. For example in a certain population of humans there will be higher percentage of people nearer the most common height and less people at either extremes.

The visual data will be analysed to determine if, how and when each of the previously predicted phenomena happen. The visual data will be recorded using a mix of screen shots of the simulation as well as video recordings. The screenshots will then be annotated and fully analysed in this document showing any visible phenomenon with details on how it may have occurred. In order to reduces the sheer quantity of data that will be collected only a selection of the visual data of the test runs will be recorded as a video where as the rest will be collected as a selection of screenshots.

4 - Implementation:

Although all of the ideas that are needed to be implemented have been designed already there was still a large amount of work implementing these ideas. Due to the mixture of pure C++ coding and the Unreal Engine's flow chart language where certain components should been implemented had to be decided.

4.1 - Agents

The agents were implemented using both C++ code as well as the inbuilt flow chart programming language called Blueprint. The C++ code was used to implement the base genotype of the agents as well as the species check. The SpeciesPawn C++ class inherited from the DefaultPawn class in Unreal Engine 4 which contained all the basic movement and mesh integration. Inheriting from this base class allowed for the SpeciesPawn class to be more focused on the evolution and interactions rather than the integration of basic assets. From the SpeciesPawn C++ class a Blueprint class was derived. Within this Blueprint class all the phenotype interpretation occurred as well as all of the checks to determine death, mating and attacking.

The interactions and decision making side of the agents was implemented in a behaviour tree. As seen in Figure one, the decisions are broken down into the need for food and the need to breed. The most important of these two needs is the need to eat and as such will override then need to mate.

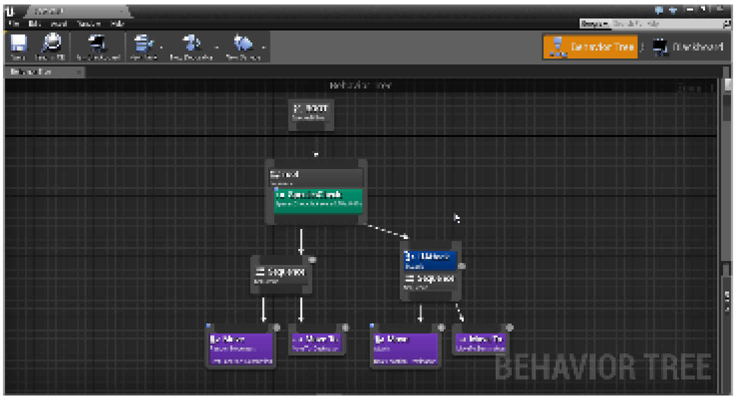


Fig 1. Species pawn Behaviour tree

Each time the behaviour tree is looped, the agent makes a radial check to determine how many agents of the same species and how many agents of a different species are within the area around itself. This process can be seen in Figure two. Each of the agents found both same species and not will be used later in the decision of where the agent will move.

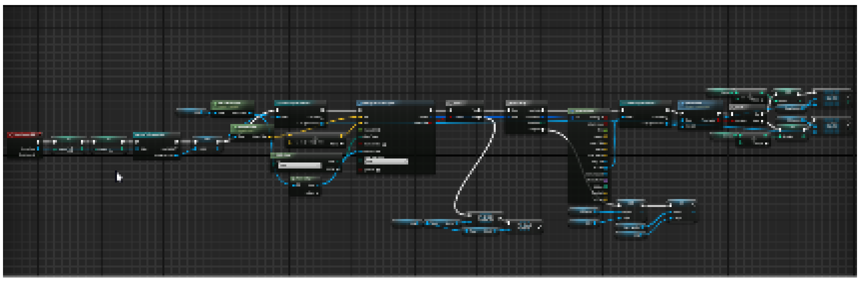


Fig 2. Species Check

If the agents does not decide that it has the need to eat, then by default it will use the method displayed in Figure 3. This method will either provide the agent with a random reachable point around it or a random reachable point around a viable mate. Which of the previous random reachable point is dependant on if the agent is able to mate or not.

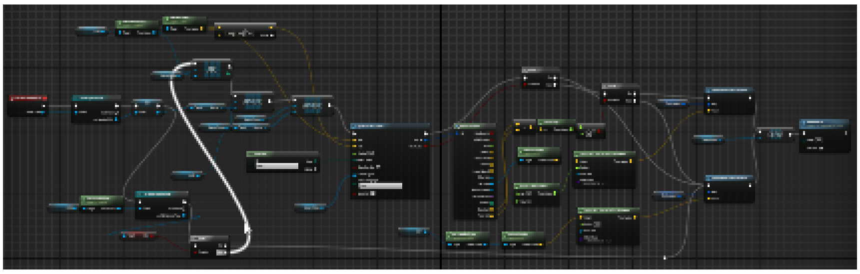


Fig 3. Mating or Random movement

The other path the agents will take if they have the need to eat. This can be seen in Figure four. Built much the same as the Mating and random movement method this method gets the point of the closest agent of a different species and sets that as the goal for the agent to move to.

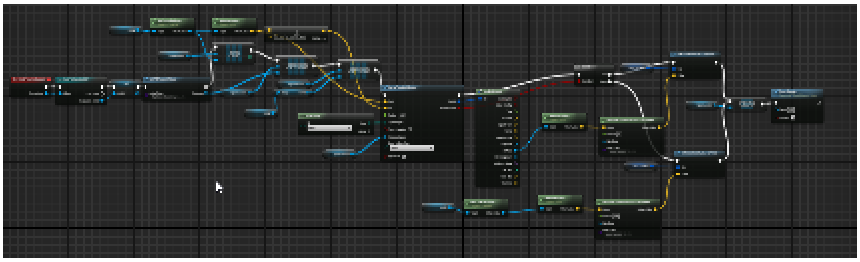


Fig 3. Attacking movement

4.2 - Evolution

Evolution is this simulation will be governed by an inbuilt inheritance method that all the agents have. This method will be implemented within the SpeciesPawn C++ class. Since there will be two different times where an agent will be initialised with values there will be two different methods. The first of the methods will be for the first generation. Since the first generation will not have any parents to inherit from this initialisation method either set a value as a predetermined value or given a random value within a predetermined range. An example of a predetermined value is the SVA which is set at a value that should ensure that there is a high likelihood of a next generation being created preventing the simulation from ending without any evolution occurring. Comparatively for every agent other than the first generation a different inheritance method is required. This different method will involve choosing a value from two parents either by random or for a specific reason. One example is that the size value is taken from the larger of the parents. However in the case of all of the chosen values a predetermined randomness is added to simulate mutation.

4.3 - GUI Elements

The GUI elements were implemented using the Unreal widget tools. The Unreal widget tools provided a great platform to create user friendly GUI elements with simple button objects, text fields and panels. With the use of these simple components very user friendly panels were created to incorporate all the required information that needed to be displayed to the user. An example of a GUI element can be seen bellow in Figure four.

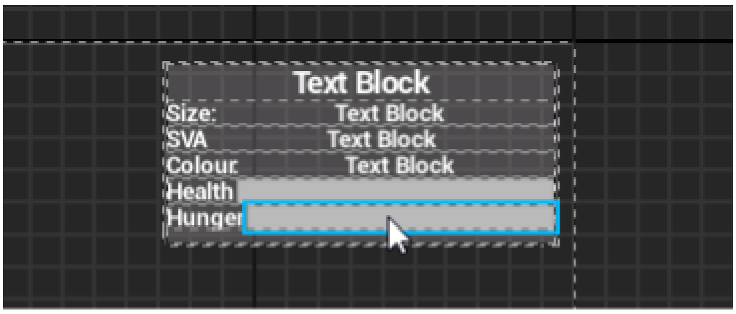


Fig 4. Agent Details Panel

4.4 - Data Saving

Although it is thought to be a simple task of taking genotype values and exporting these as a text file the implementation of saving data was much harder than originally thought at the design stage. In order to export data as a text file a C++ class extending the Blueprints function library had to be created. This C++ class contained two methods the first to export a single string as a .txt file and the second to take a whole array of strings and save as a single .txt file. These methods will help to export the data from within the centralised level class that dealt with all of the simulations running. However the data still had to be extracted from each of the agent objects and changed into a single string to be passed to the saving method. This was achieved within the level blueprint by simple get methods and concatenating the data from each agent into a string. This method would create a single string for each agent within the simulation and as such a further step is needed. This final step is to take all of the strings for each of the agents and add them to a single array which will be passed to the previously created saving method.

5 - Testing:

Testing showed that a selection of problems would occur occasionally which would create inconsistencies in the results. Although not often, these inconsistencies meant that for that specific test run the data would create many outliers. One such problem would be that, since the starting population was randomly initialised, there was a possibility that none of the agents would spawn anywhere near another of its species. Since no viable mates were within reach no reproducing populations would occur and so all of the agents would die out. To decrease the chance of this problem occurring the size of the starting population was increased. Another problem occurred when the entire population of agents got to more than two hundred agents performance issues occurred and the simulation stopped running. This issue was solved by adding a culling mechanism which would remove the agents of the lowest health from the population. Due to the problems previously stated the length of each simulation was vastly different and so a cut off point was chosen to help keep the data standardised.

5.1 - Visual Testing

Visual testing proved to be a hard and tedious task since there was a chance that multiple populations of multiple species could appear in different areas in the simulation. The fact that there could be phenomena occurring anywhere during the simulation made it hard to ensure that every important phenomenon was viewed and recorded. However hard it was to keep track of everything that was happening during the test runs it was still possible to get some very useful data.

5.2 - Genotype Data

Since the process of collecting this form of data was automated there were no problems faced other than the general ones discussed previously (i.e. the exportation of the data). The difficulties with this form of data came when needing to analyse it. Since there was such a large quantity of data (on average one hundred agents with five floating point values every five seconds for more than two hundred seconds) the processing of data from base text to the graphs that will be analysed.

6 - Results:

Using the test cases designed in a previous chapter the results of which will be discussed and compared to the predictions made at the start of this report. Furthermore any problems or anomalies seen in the results will be discussed and the possible reasons for these anomalies will be predicted.

6.1 - Visual Results

The results in this section are what has been recorded with the visual test cases and as such all the phenomena that should be seen by visual testing will be analysed and evaluated.

Proof of Flocking

Flocking was found to be the most common of the phenomena and in every single test run there was almost continual evidence of flocking occurring. The only times where flocking was not visible was at the beginning of the simulation where there was a chance that the agents were not near any agents of the same species. However this would also occur at other times during the simulation due to the same requirement being present.

Proof of Divergence

Divergence was proved to occur many times over all of the runs as was far more visible than predicted. The figure below shows an example of this phenomenon occurring during test run two. From the first picture there is a clear cluster of a single species even though there are also many other agents scattered around this cluster the cluster itself is still prominent. Moving through the pictures the split from the single more orange yellow species to the branched yellow green species as well as the first species. Furthermore on the final picture another branched species can be seen in the bottom left more in the turquoise spectrum of hue. This progression and branching out of a single species clearly shows that divergences is occurring.

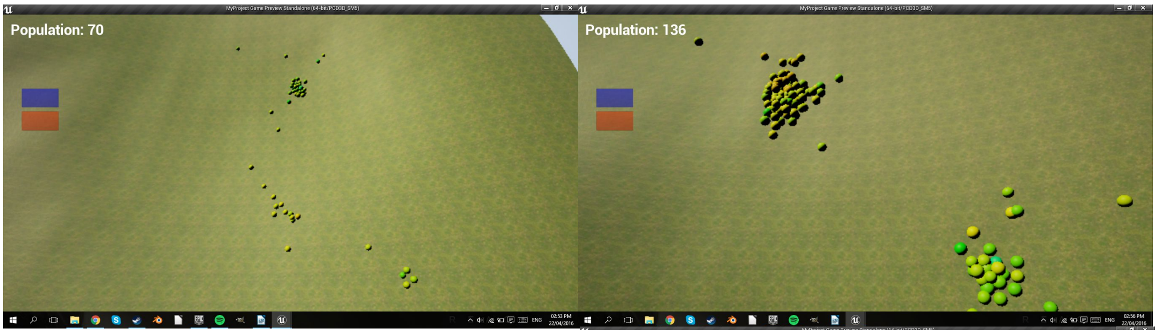


Fig 5. (left to Right, Top to Bottom) A single species branching into two sub-species

<u>Proof of Convergence</u>

Convergence has been seen in multiple test runs of the simulation however not as frequently as divergence due to the requirement of more than one species. In the below example there are three visible species the top purple, the bottom turquoise and the middle dark blue. The dark blue species has been created through the convergence of a branch off of these two other species. Although all the middle population is product has ancestors from the two other populations none of the populations consider the others as the same species. This change into three separate species has occurred through the first convergence followed by continued evolution away from the starting populations.

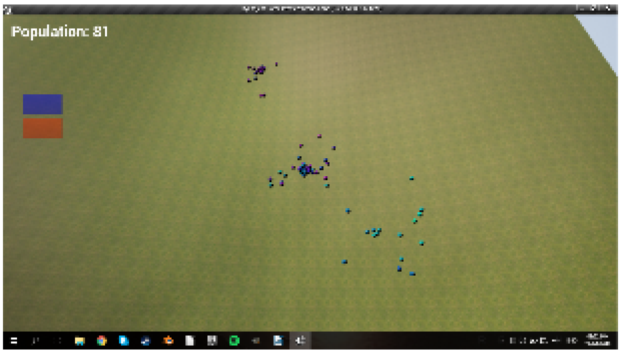


Fig 6. Three visible populations turquoise, purple and the convergence of the two in the center which is blue

Proof of Predator-Prey

Predator-prey dynamic, although not as frequently seen as the previous phenomena was seen more often than was predicted. This phenomenon most often occurred when two different species occupied the same space. The Figure seven shows an example of proof of this phenomenon from test run three. As the figure shows there is two different shades of blue, turquoise and dark blue, which are within the same space. These two species cannot live in the same area without attempting to eat each other and as such must be showing predator-prey behaviour.

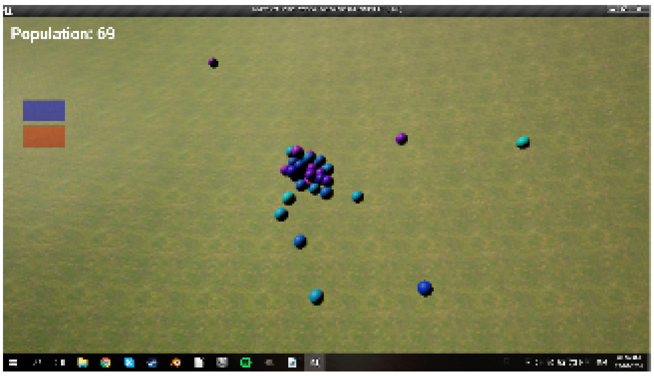


Fig 7. Two different species occupying the same space causing predator prey behavior

6.2 - Genotype Data Results

From the large amount of data collected from all of the test runs that were completed a great many of predictions can be seen as correct and many conclusions can be made. Although as previously stated about the test cases there were a small amount of inconsistencies the majority of the results pointed to the same conclusions that will be discussed and made clear below.

Normal Distribution Curves

The graph below shows the normal distribution curve can be seen in populations of agents within the simulation. Although the curve is visibly different to the iconic bell shaped curve seen in real populations this graph is very close and shows how close the simulation is to real. This proves that the simulation is at the very least capable of creating realistic population distribution.

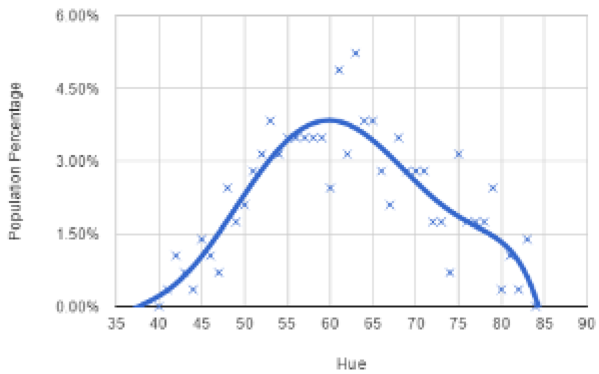


Fig 8. Graph showing the normal distribution of a population's Hue

Migrating populations and specific species

The graph below shows that over time the whole population of agents in the simulation has changed from a very varied group of species to a single specific species. In this test case only one species has developed and survived from the many that were present at the start. Furthermore it is also visible in this graph that the populations hue changes over time which shows the population is migrating through the problem space.

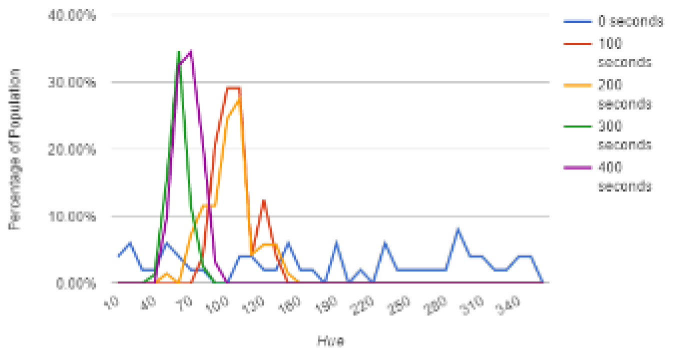


Fig 9. Graph showing the migration of populations and speciation

Further proof of this can be seen in the graph below.

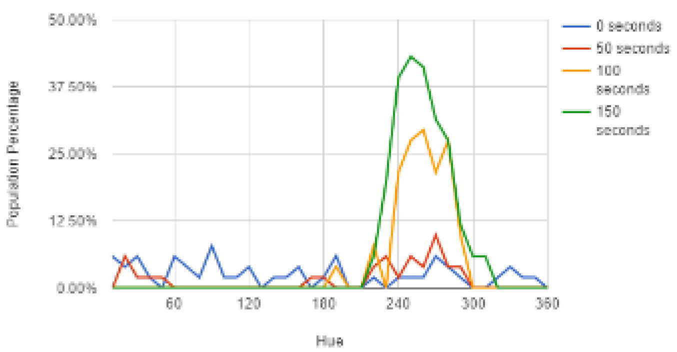


Fig 9. Another graph showing species migration and speciation

Self Selected Traits

Self selected traits is a phenomenon which was harder to record than the others before. Since the benefits of being a smaller size were not huge, in general the agents evolved into larger and large sizes. Due to this issue the visible phenomenon of selection is simpler than previously hoped for. Although the general correlation shows an increase in size there is however increases and decreases in the variation of size. These variation show branching sub species that have failed to survive long enough to be considered a species of their own.

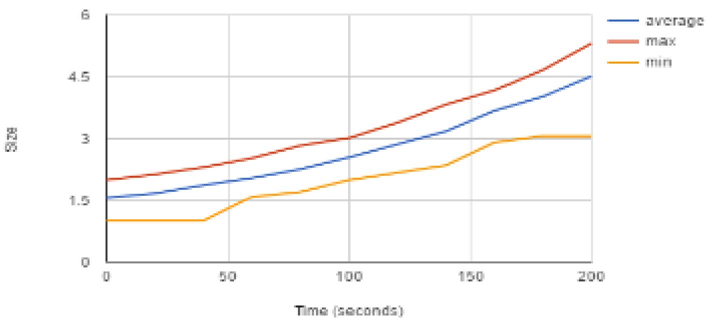


Fig 11. Graph showing the progression of population size

The change of species variance acceptance emulate a similar pattern as the population size. However, from Figure twelve below, the variation with the SVA is much higher than that of the population size in the graph above. Even though the general trend in this graph shows an increase in SVA values it is not as strong of a correlation as that in the previous example. Furthermore there are points in the simulation where the average does decrease unlike in the size graph. The fact that the SVA both increases and decreases can be said to be proof of self selected traits within the populations.

However this single graph does not show significant enough values to confirm this suspicion and further data is required.

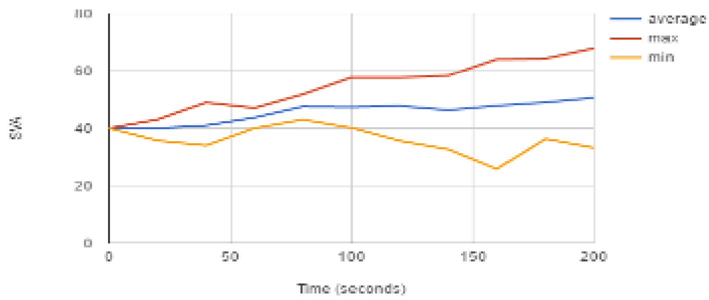


Fig 12. Graph showing the progression of SVA values

Figure thirteen provides more evidence of this oscillating average value of SVA which give proof of self selected traits. As the graph shows that at different points during the simulation the SVA values increased and decreased. This changing average could be explained by the ideas that the species are faced by the need to be less or more specific. This need for changing the SVA could be to allow interbreeding with other species to ensure survival. Alternatively this correlation could simple have occurred through random mutation

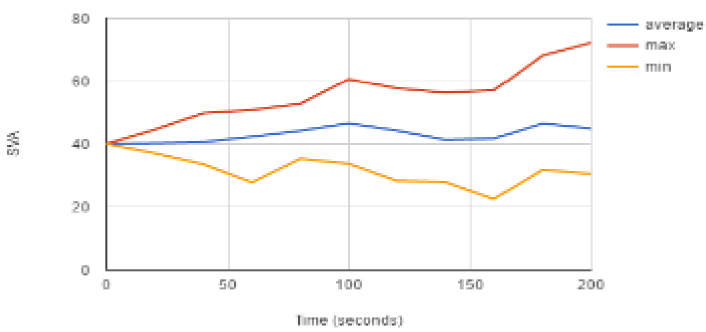


Fig 13. Graph showing the progression of SVA values

7 - Evaluation:

In terms of results the quantity of the gathered results was a good amount. However if more time was allocated to testing a larger data set would have been collected. Furthermore tweaking of the simulation should have been done to help allow for longer run times of the test cases. These longer tests would have helped to further analyse the phenomena and ensure that the collected results were not just what happened in the early part of the simulation. Although not as important as possibly adding more features this addition would have helped to cement the success of this project. On a separate note user testing of individuals would have been preferable. This user testing would have helped to determine how effective this simulation is for teaching people the ideas of evolution. Although the main aims of this project were to determine if certain real world phenomena could be simulated in the proposed methods the use of this simulation as an aid in explaining evolution was also proposed as a possible aim.

8 - Conclusion:

Comparing the end result of this project to the predicted result, hopes and aims it is possible to see that in many areas this project has been a success. All of the predicted phenomena stated at the start of this project were seen either in every single test run of the simulation or seen enough time to not be considered anomalies. Furthermore the quantitative data showed some very strong correlations and similarities towards real life populations. This shows that the ideas and designs created in order to recreate this phenomena were correctly theorised and then implemented.

In terms of implementation of this project there was elements of success as well as certain elements that would require more improvement or Most of the problems faced in this project were the result of learning and using the unreal engine 4 platform. . Although many problems came about due to unreal engine 4 it also help reduce the complexity of certain areas of this project which if attempted with another software would have made this project take a much longer time and would possibly not have been completed as much as it was. Looking back at the difficulties faced during this project it would have been beneficial to have completed a study to find if there were any better choices in technologies that could have been used.

Finally altogether even without the implementation of the third and final version of the designed agents this project has been successful in implementing an evolution system that does not require an evolutionary algorithm to correctly simulated realistic evolution. Speculating on the impact that the final version of the agents would have on the simulation, it is possible to expect that the similarities with the simulation to real world phenomena would be much higher than the end result that this project had.

9 - Further Work:

Following the ideas created in this project possible further work could be done to build on top of the already completed work. One such additions that could be made would be to add a more dynamic and constantly changing environment that would better simulate the real world. This improved simulated environment could include aspects such as temperature change, climate and seasons. The addition of these new environmental aspects would greatly improve the realism and validity of this simulation. This is due to the way that people and animals have evolved within different climates and therefore the simulation should allow the development and continuation of certain traits dependant on the environment the species will be in and therefore will be more realistic. It is possible to predict that these additions would also help to encourage the occurrence of divergent evolution as per the definition that is provided in Chapter one.

Another component that could be added in further development would the addition of different genders within the populations. By adding in different genders the mating function within the simulation will develop to reflect a real life population. This will create more interesting dynamics within species as well as the possibility of imitations of real world breeding phenomena occurring.

An improved simulation would also further increase its usefulness as a teaching aid as it would be a far more impressive and reliable show of real world evolution. Thus allowing its educational use to be increased as the concepts and theories within evolution will be better presented and displayed within the simulation. The further developments that can be implemented on this basic simulation will allow further complex ideologies and theories to be shown and therefore allows its educational use to be developed.

Bibliography:

Costa, L. da F., Oliveira, O.N., Travieso, G., Rodrigues, F.A., Villas Boas, P.R., Antiqueira, L., Viana, M.P. and Correa Rocha, L.E. (2011) ‘Analyzing and modeling real-world phenomena with complex networks: A survey of applications’, Advances in Physics, 60(3), pp. 329–412.

Darwin, C. (2008) *On the origin of species*. Edited by Gillian Beer. New York: Oxford University Press.

Dieckmann, U., Doebeli, M., Metz, J.A.J. and Tautz, D.(eds), 2004. Adaptive Speciation. Cambridge University Press

Harvey, I., 1992. Species adaptation genetic algorithms: A basis for a continuing SAGA. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (p. 346-354).

Kuang, Y., Beretta, E., 1998. Global qualitative analysis of a ratio-dependent predator–prey system. Journal of Mathematical Biology, 36(4), p.389-406

Lack, D. (1983) *Darwin’s Finches*. United Kingdom: Cambridge University Press.

Losos, J.B., 2011. Convergence, Adaptation, and Constraint. Evolution, 65(7), p. 1827-1840

Martin, E. and Hine, R., 2008. A Dictionary of Biology. 6th ed.Oxford: Oxford University Press

Miconi, T. and Channon A., "A Virtual Creatures Model for Studies in Artificial Evolution," in Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005), Edinburgh, Sims. K. Evolving 3d morphology and behavior by competition. ALife IV : Proceedings of the 4th Conference on Artificial Life, pages 28-39. MIT Press, 1994.

Stanton, A. and Channon, A.D., Incremental Neuroevolution of Reactive and Deliberative 3D Agents. In Advances in Artificial Life, ECAL 2015: Proceedings of the Thirteenth European Conference on the Synthesis and Simulation of Living Systems. Keele University.Deliberative 3D Agents, in Proceedings of the European Conference on Artificial Life 2015 (ECAL 2015), p. 341-348, MIT Press, 2015

Stern, D.L., 2013. The genetic causes of convergent evolution. Nature Reviews Genetics, 14, p. 751-764

Strickberger, M.W., Hall, B.K. and Hallgrimsson, B. (2007) *Strickberger's evolution*. 4th edn. United States: Jones and Bartlett Publishers.

School of Computing and Mathematics: Student Project Ethics Committee Application form (U/G and PGT Students)

Please print off a hard copy of this form and submit it to the School Office.
Please remember to sign it, date it, and to get your supervisor's signature on it.

Student name:	**Tom Frederick Turton**
Student number:	13013056
Course:	Bsc. Single honours Computer Science
Date:	11/11/2015

Part A: (all students)

Does the topic of the project involve any of the following?	**YES**	**NO**
Recall of personal or sensitive memories		X
Reporting or discussion of personal or sensitive topics		X

Tasks which could be harmful or distressing		X
A significant risk of participants later regretting taking part		X
Procedures which are likely to provoke inter-personal or inter-group conflict?		X

If you answer "Yes" to any questions on Part A, then please also complete the University Ethics form (on the KLE) and seek guidance from the School Research Governance officer. The School Research Governance officer (mentioned in the university form) is the projects co-ordinator, Gordon Rugg.

Part B: (if you are doing a software design and/or software build or gathering data from human participants)

Requirements gathering and evaluation: use of unusual techniques	YES	NO
Will the techniques you are using for requirements gathering and software evaluation be unusual in a way which could cause ethical problems?		X
Will any of the participants be from a vulnerable group (e.g. under 18, or with learning difficulties, or under pressure to help you)?		X

If you answer “Yes” to any questions on Part B, then please also complete the University Ethics form (on the KLE) and seek guidance from the School Research Governance officer. The School Research Governance officer (mentioned in the university form) is the projects co-ordinator, Gordon Rugg.

Student’s signature and date:	
Supervisor’s signature and date:	

<u>UG Project Plan</u>
<u>CSC-30013/14</u>

Project Overview and Description

Student Name: Tom Frederick Turton
Student Username: w0f90
Student Number: 13013056
Module (*delete as appropriate*): CSC-30014
Supervisor Name: Alastair Channon
Project Title: The use of evolutionary neural networks to simulate animal behaviour

Please provide a brief Project Description:

I will be using an evolutionary neural network in an attempt to create animals with intelligence to be used in a game environment. The hope is to create an environment that will be seemingly random to the player and more realistic instead of the more often used form of AI in games which is scripted behaviour.

What are the aims and objectives of the Project?

My aims are to create intelligent animals that choose how they move around in the game environment. Furthermore I wish to allow the player to have a certain level of control over the environment such as adding new animals. My project can be split into three main objectives. Firstly the creation of animal 3d models and animations. Secondly the creation and evolving of nerual networks that exhibit the animal behaviours I require. Lastly making a GUI for the player to interact with the game environment.

Please provide a brief overview of the key literature related to the Project:

The main paper[1] I am using to relate to my project is in junction with the NERO artificial intelligence game. Other papers involve talking about the general use of genetic neural networks in game AI through example games and situations such as Pac-Man, the iterated prisoner's dilemma and checkers.

Project Process and Method

Please provide a brief overview of the Methodology to be used in the Project (inc. an overview of best practice within the Methodology):

For my methodology I will be using a prototyping method in general but within each prototype I will be using a waterfall methodology of finishing each section before moving on to the next. I will be verifying the results of this project by counting both the number of unique interactions and behaviours the each animal exhibits with in the game to determine if the project is a success or failure.

Which Data Collection Methods will be employed (e.g card sorts, questionnaires, simulations, …)?

The only interaction with users I will have in this project is for a group of people to test the GUI and ensure that it does not have any problems with it. Furthermore I will have to collect data on animals that I will have in my game to ensure that my simulation is as close to real life as I can make it.

Time and Resource Planning

Will Standard Departmental Hardware be used? YES/NO

If NO please outline the Hardware/Materials to be used:

YES

Will Software which is already available in department be used? YES/NO

If NO please outline the Software to be used including how any necessary licences will be obtained:

NO, Either the Unreal engine 4 will be required or the OpenNERO game engine.

Will the project require any Programming? YES/NO

If YES please list the (potential) Programming Languages to be used (including any IDEs and Libraries you may make use of):

YES, C++

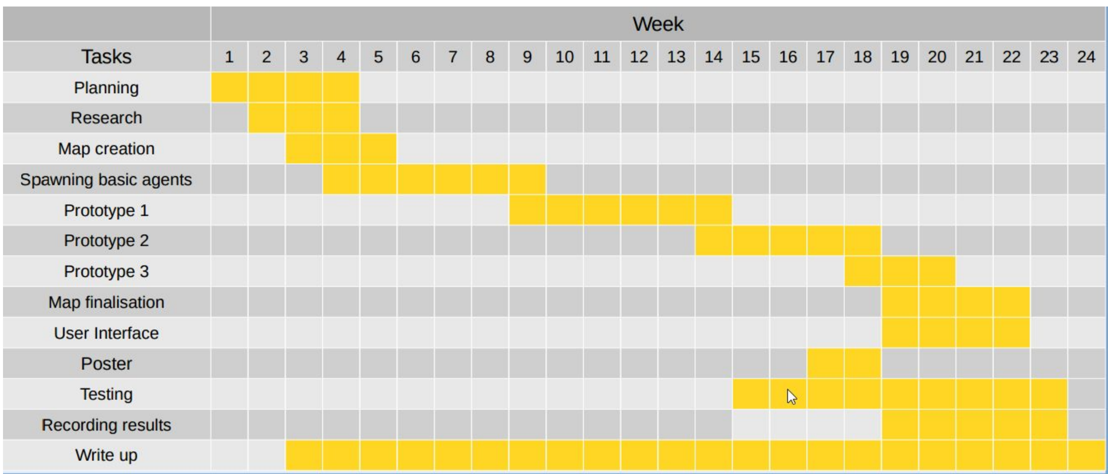
Table of Risks (*if non Standard Hardware and/or Software to be used please include backup options/ contingency plans here)*:

Risk	**Description**	**Actions to take**	**Method to prevent**
Updates from the software having an effect	New updates on the software I use (unreal 4 for example) could cause	Spend time to bring my project up to date to work	I will have multiple versions of the project and

on work already finished	my project to not work the way it did on an older version.	with the new update.	when a new update is release I will make a new version and test to see if there are changes.
No users available to test my software	No one is free to test my software.	I will have to deal with testing the software myself and deal with the lack of feedback.	I will organised times to test my software with plenty of time to ensure no problems occur.
Loss of data	Loosing a hard drive or files being corrupted.	Revert to an old save and try and bring it back to	I will have multiple copies of my work both on

		what it was.	the cloud and on hard drives.
Project not being completed	Loosing track of time or not predicting the time taken to complete tasks correctly causing the project to not finish.	Wrap up what has been completed and explain the problems faced with completing the project in the report.	Through the planning section I will ensure that the time constraint is kept to to prevent running out of time.

Gantt Chart/ Pert Chart (must include milestones and deliverables):



References and Administration

Please include a list of References used in this Plan:

[1]Stanley, KO. Bryant, BD. Miikkulainen, R.(2005) Evolving
Neural Network Agents in the NERO Video Game,
ftp://www.cs.utexas.edu/pub/neural-nets/papers/stanley.cig05.pdf

[2]http://users.auth.gr/~kehagiat/Research/GameTheory/12CombBiblio/GamesEvolutionNeuralNets.pdf

[3]http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.6629&rep=rep1&type=pdf

Submission Date:

PLEASE NOTE THAT SHOULD YOUR PROJECT UNDERGO ANY MAJOR CHANGES FOLLOWING THE SUBMISSION OF THIS PLAN YOU ARE EXPECTED TO SUBMIT AN UPDATED PLAN WHICH ACCURATLEY REFLECTS YOUR PROJECT.

CHANGES IN MODULE FROM CSC-30013 TO CSC-30014 ARE DEEMED TO BE A SIGNIFICANT CHANGE REQUIRING AN UPDATED PLAN.